

単語定義による Web アプリケーション自動生成システムの提案

高橋克弥† 水本重幸† 山田敬三† 田中充† 佐々木淳† 船生豊†

†岩手県立大学ソフトウェア情報学部

1 はじめに

昨今の情報システム開発では、短納期やコスト削減という要求から、より効率的に情報システムを開発することが求められている。現在では、プログラムの自動生成ツールやパッケージを利用した開発が行なわれることが多い。しかし、開発案件毎にツールの選定や学習が必要であり、システムによってはカスタマイズに多大な労力がかかる場合もあり、効率的にシステムを開発できているとはいえない。

我々は、近年発展が著しい Web アプリケーションを対象とし、開発案件に依存しない統一的な方法による新しい情報システム開発手法の実現を目指している。本稿は、Web アプリケーションにおいて入出力されるべきデータを単語として定義し、その単語から Web アプリケーションを自動生成するシステムを提案する。

2 既存研究

Web アプリケーションの開発は View ベースとモデルベースのアプローチがある。View ベースのアプローチについては、Web アプリスケルトン[1]の研究がある。この Web アプリスケルトンとは、ユーザによって作成された Web サイトからページの類似度を測定し、テンプレートを作成するものである。Web アプリの雛型は作成できるがロジックについては生成されていない。一方モデルベースの研究例として、八木橋ら[2]は、根来氏が生み出した Lyee 理論を用いて、単語別プログラムによる情報システムの構造確立について科学的考察を行っている。ここでいう「単語」とは、値を格納する単一のデータ項目とその値を決定するための処理を組にしたものであり、システムを構成する最小単位の部品である。この理論では、ソフトウェアの開発は、単語別プログラムとロジックを作り出すためのイテレーションから成り立っているとされている。単語別プログラムは、システムのプログラム部分を単語レベルまで細かく分割した形をとっている。単語の

設計については画面への出力部から考え、それを表示するための定義式を設定する。定義式を実行するために必要な式を単語として定義していくことを繰り返し、最終的に外部入力や DB からの入出力へと定義がされていく。イテレーションでは、各単語別プログラムを順次実行し、値が決定していくまで繰り返すことにより、制御ロジックを作り上げる。また、その理論を実現する手段として、プロセス代数をベースとした言語が提案されている[3]。しかし、使用するにはその代数理論を理解する必要があり、その難解さからまだ一般には普及していない。特に Web アプリケーションを対象とした研究例は少ない。

3 提案内容

3.1 単語定義プログラミング言語

我々は、開発者に馴染み易いプログラムベースで単語を定義するための新しいプログラミング言語を開発することとした。この言語は、一般的なプログラム言語にある四則演算や大小比較、制御処理を定義式として表現でき、単語名とその単語の値を決定するための定義式を記述するという形をとる。開発者は、画面ごとに必要となると思われる単語を列挙し、前述の言語を用いて単語定義する。

3.2 Web アプリケーション自動生成システム

我々が提案する単語定義からの Web アプリケーション自動生成については以下の通りである。

開発者は以下の 3 点を準備する。

- 単語定義書
- 画面設計書 (html ファイル)
- DB 設計書

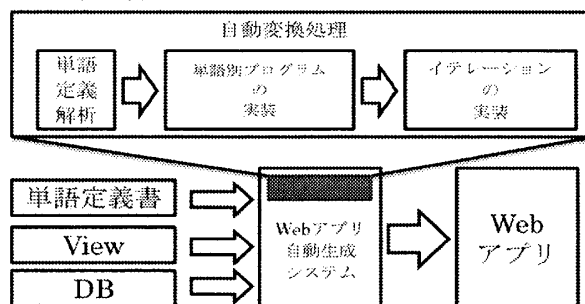


図 1. 自動生成システムイメージ図

Proposal of Web Application Automatic Generation System by Word Definition
Katsuya TAKAHASHI†, Shigeyuki MIZUMOTO†, Keizo YAMADA†,
Michiru TANAKA†, Jun SASAKI†, Yutaka FUNYU†
†Faculty of Software and Information Science, Iwate Prefectural University

自動生成システムでは、単語定義書から既存のプログラム言語に変換を行う（本稿では PHP への変換を想定している）。この変換によってプログラムに単語別プログラムとイテレーションの仕組みを導入する。本提案システムの全体の流れを図 1 に示す。

3. 3 自動変換処理

単語定義書から既存プログラム言語への変換処理の内容としては、以下の内容となる。

(1) 単語定義解析

単語定義解析では、JavaCC によって作ったコンパイラを用い、単語定義書が書式通りに書かれているかのチェックを行う。解析に問題がなければプログラムの自動生成が実行される。

(2) 単語別プログラムクラスの実装

単語定義書から解析して得られた単語情報により単語ごとのクラスを生成し、単語別プログラムクラスを継承させ、単語の名前のセット、定義式の登録を行うための処理を自動生成する。ここでいう単語別プログラムクラスは、単語の名前、定義式、値の三つの領域を定義し、機能として単語名、定義式、値の登録や値が登録されているかを判定するためのフラグをシステム側で実装している。クラス図の例を図 2(左)に示す。

(3) イテレーションクラスの実装

イテレーションクラスを生成し、上記自動生成された単語別プログラムをイテレーションクラス内にある配列に格納するための処理を自動生成する。ここでいうイテレーションクラスは、画面名、単語を管理する配列の領域を定義し、機能としてイテレーション内の単語別プログラムを繰り返し実行することにより制御ロジックを実装するように作られている。イテレーション内の単語別プログラムに変化がなくなったときに入力待ちまたは出力状態となるようにシステム側で実装している。クラス図の例を図 2(右)に示す。

subject	iteration
\$name \$value \$defined	\$name \$defined \$palette
getName() setName(\$name) getValue() setValue(\$value) isDefined() setDefined(\$flag) iteration() secondBox()	getName() setName(\$name) isDefined() setDefined(\$flag) getSubjectArray() getSubject(\$name) iteration()

図 2. 左：単語別プログラムクラス
右：イテレーションクラス

最後にイテレーションクラスを実行させるための処理を自動生成する。

上記 3 つの処理により Web アプリケーションで稼働可能となる処理は自動生成される。ユーザと開発者は、実際に稼働している Web アプリケーションを見て、変更部分の検討を行う。変更がある場合単語定義書の作成からやり直す。

4 考察

今回、Web アプリケーション自動生成システムを構築する第 1 ステップとして、生成されると予想されるプログラムコードを手書きで作成し、動作することを確認した。今後は、上記プログラムコードを自動生成するための変換処理機能を定義していく。また、Web アプリケーション自動生成システムにおいて単語定義書では表現できない処理をコンポーネントとして定義し、開発者に利用できるようにする予定である。また、多言語化への検討も行っている。開発者とユーザは協同で、単語定義書と画面設計書、DB 設計書を作成し、その設計書を本システムに入力することにより即座に Web アプリケーションを得ることが可能である。単語ごとのプログラム設計により後の仕様変更にも対応しやすいうという保守性の面での効果の他によりユーザの要求を反映したシステム開発が可能になると考えられる。

5 まとめ

本稿では、Web アプリケーションを対象とした統一的な情報システム開発手法を実現するために単語定義言語と自動生成システムの提案を行った。

今後、通常のプログラミングによる開発と提案システムを利用した方式での開発との比較実験を行い、有効性を検証していく予定である。

参考文献

- [1]Sota Honda, A Collaborative Environment for User-initiated Development of Web Applications, SoMeT'07, pp.494-504, 2007
- [2]Hideo Yagihashi, Scientific Research into Establishment of Lye's Word-based Program Structure, SCI2001, Volume XVI, pp238-245(2001)
- [3]B.Ktari, From Lye-Calculus to Java Code, SoMeT'04, pp.283-300, 2004